Jon Home

THE CRONUS VIRTUAL LOCAL NETWORK

William I. MacGregor
Daniel C. Tappan
Bolt Beranek and Newman Inc.

25 August 1982

[The purpose of this note is to describe the CRONUS Virtual Local Network, especially the addressing related features. These features include a method for mapping between Internet Addresses and Local Network addresses. This is a topic of current concern in the ARPA Internet community. This note is intended to stimulate discussion. This is not a specification of an Internet Standard.]

1 Purpose and Scope

This note defines the Cronus (1) Virtual Local Network (VLN), a facility which provides interhost message transport to the Cronus Distributed Operating System. The VLN consists of a 'client interface specification' and an 'implementation'; the client interface is expected to be available on every Cronus host. Client processes can send and receive datagrams using specific, broadcast, or multicast addressing as defined in the interface specification.

---

(1) The Cronus Distributed Operating System is being designed by Bolt Beranek and Newman Inc., as a component of the Distributed Systems Technology Program sponsored by Rome Air Development Center. This work is supported by the DOS Design/Implementation contract, F30602-81-C-0132.

From the viewpoint of other Cronus system software and application programs, the VLN stands in place of a direct interface to the physical local network (PLN). This additional level of abstraction is defined to meet two major system objectives:

- COMPATIBILITY. The VLN defines a communication facility which is compatible with the Internet Protocol (IP) developed by DARPA; by implication the VLN is compatible with higher-level protocols such as the Transmission Control Protocol (TCP) based on IP.

- SUBSTITUTABILITY. Cronus software built above the VLN is dependent only upon the VLN interface and not its implementation. It is possible to substitute one physical local network for another in the VLN implementation, provided that the VLN interface semantics are maintained.

(This note assumes the reader is familiar with the concepts and terminology of the DARPA Internet Program; reference [6] is a compilation of the important protocol specifications and other documents. Documents in [6] of special significance here are [5] and [4].)

The compatibility goal is motivated by factors relating to the Cronus design and its development environment. A large body of software has evolved, and continues to evolve, in the internet community fostered by DARPA. For example, the compatibility goal permits the Cronus design to assimilate existing software components providing electronic mail, remote terminal access, and file transfer in a straightforward manner. In addition to the roles of such services in the Cronus system, they are needed as support for the design and development process. The prototype Cronus cluster, called the Advanced Development Model (ADM), will be connected to the ARPANET, and it is important that the ADM conform to the standards and conventions of the DARPA internet community.

The substitutability goal reflects the belief that different instances of the Cronus cluster will utilize different physical local networks. Substitution may be desirable for reasons of cost, performance, or other properties of the physical local network such as mechanical and electrical ruggedness. The existence of the VLN interface definition suggests a procedure for physical local network substitution, namely, re-implementation of the VLN interface on each Cronus host. The implementations will be functionally equivalent but can be expected to differ along dimensions not specified by the VLN interface definition. Since different physical local networks are often quite similar, the task of "re-implementing" the VLN is probably much less difficult than building the first implementation; small modifications to an existing, exemplary implementation may suffice.

The concepts of the Cronus VLN, and in particular the VLN implementation based on Ethernet described in Section 4, have significance beyond their application in the Cronus system. Many organizations are now beginning to install local networks and immediately confront the compatibility issue. For a number of universities, for example, the compatibility problem is precisely the interoperability of the Ethernet and the DARPA internet. Although perhaps less immediate, the substitutability issue will also be faced by other organizations as local network technology advances, and the transfer of existing system and application software to a new physical local network base becomes an economic necessity.

Figure 1 shows the position of the VLN in the lowest layers of the Cronus protocol hierarchy. The VLN interface specification given in the next section is actually a meta-specification, like the specifications of IP and TCP, in that the programming details of the interface are host-dependent and unspecified. The precise representation of the VLN data structures and operations can be expected to vary from machine to machine, but the functional capabilities of the interface are the same regardless of the host.

```
                    .
                    .
|                   .                  |
|--------------------------------------|
| Transmission   |  User      |        |
| Control        |  Datagram  |  ...   |
| Protocol       |  Protocol  |        |
|--------------------------------------|
|           Internet Protocol          |
|                 (IP)                 |
|--------------------------------------|
|         Virtual Local Network        |
|                (VLN)                 |
|--------------------------------------|
|         Physical Local Network       |
|         (PLN, e.g. Ethernet)         |
 --------------------------------------
```

Figure 1 . Cronus Protocol Layering

The VLN is completely compatible with the Internet Protocol as defined in [5], i.e., no changes or extensions to IP are required to implement IP above the VLN. In fact, this was a requirement on the VLN design; a consequence was the timely completion of the VLN design and avoidance of the lengthy delays which often accompany attempts to change or extend a widely-accepted standard.

The following sections define the VLN client interface and illustrate how the VLN implementation might be organized for an Ethernet PLN.

2 The VLN-to-Client Interface

The VLN layer provides a datagram transport service among hosts in a Cronus 'cluster', and between these hosts and other hosts in the DARPA internet. The hosts belonging to a cluster are directly attached to the same physical local network, but the VLN hides the peculiarities of the PLN from other Cronus software. Communication with hosts outside the cluster is achieved through some number of 'internet gateways', shown in Figure 2, connected to the cluster. The VLN layer is responsible for routing datagrams to a gateway if they are addressed to hosts outside the cluster, and for delivering incoming datagrams to the appropriate VLN host. A VLN is viewed as a network in the internet, and thus has an internet network number. (2)

---

(2) The PLN could possess its own network number, different from the network number of the VLN it implements, or the network numbers could be the same. Different numbers would complicate the gateways somewhat, but are consistent with the VLN and internet models.

```
                to internet
                 network X
                     |
                     |
       -----       -----       -----       -----
      |host1|     |gtwyA|     |host2|     |host3|
       -----       -----       -----       -----
         |           |           |           |
    ----------------------------------------------------
             |           |           |           |
           -----       -----       -----       -----
          |host4|     |host5|     |gtwyB|     |host6|
           -----       -----       -----       -----
                                     |
                                     |
                                to internet
                                 network Y
```

Figure 2 . A Virtual Local Network Cluster

The VLN interface will have one client process on each host, normally the host's IP implementation. The one "client process" may, in fact, be composed of several host processes; but the VLN layer will not distinguish among them, i.e., it performs no multiplexing/demultiplexing function. (3)

---

(3) In the Cronus system, multiplexing/demultiplexing of the datagram stream will be performed above the IP level, primarily

The structure of messages which pass through the VLN interface between client processes and the VLN implementation is identical to the structure of internet datagrams constructed in accordance with the Internet Protocol. Any representation for internet datagrams is also a satisfactory representation for VLN datagrams, and in practice this representation will vary from host to host. The VLN definition merely asserts that there is ONE well-defined representation for internet datagrams, and thus VLN datagrams, on any host supporting the VLN interface. The argument name "Datagram" in the VLN operation definitions below refers to this well-defined but host-dependent datagram representation.

The VLN guarantees that a datagram of 576 or fewer octets (i.e., the Total Length field of its internet header is less than or equal to 576) can be transferred between any two VLN clients. Larger datagrams may be transferred between some client pairs. Clients should generally avoid sending datagrams exceeding 576 octets unless there is clear need to do so, and the sender is certain that all hosts involved can process the outsize

←←←←←←←←←←←←←←←←

in conjunction with Cronus object management.

datagrams.

The representation of an VLN datagram is unconstrained by the VLN specification, and the VLN implementor has many reasonable alternatives. Perhaps the simplest representation is a contiguous block of memory locations, either passed by reference or copied across the VLN-to-client interface. It may be beneficial to represent a datagram as a linked list instead, however, in order to reduce the number of times datagram text is copied as the datagram passes through the protocol hierarchy at the sending and receiving hosts. When a message is passing down (towards the physical layer) it is successively "wrapped" by the protocol layers. Addition of the "wrapper"--header and trailer fields--can be done without copying the message text if the header and trailer can be linked into the message representation. In the particular, when an IP implementation is the client of the VLN layer a linked structure is also desirable to permit 'reassembly' of datagrams (the merger of several 'fragment' datagrams into one larger datagram) inside the IP layer without copying data repeatedly. If properly designed, one linked list structure can speed up both wrapping/unwrapping and datagram reassembly in the IP layer.

Although the structure of internet and VLN datagrams is identical, the VLN-to-client interface places its own interpretation on internet header fields, and differs from the IP-to-client interface in significant respects:

1. The VLN layer utilizes only the Source Address, Destination Address, Total Length, and Header Checksum fields in the internet datagram; other fields are accurately transmitted from the sending to the receiving client.

2. Internet datagram fragmentation and reassembly is not performed in the VLN layer, nor does the VLN layer implement any aspect of internet datagram option processing.

3. At the VLN interface, a special interpretation is placed upon the Destination Address in the internet header, which allows VLN broadcast and multicast addresses to be encoded in the internet address structure.

4. With high probability, duplicate delivery of datagrams sent between hosts on the same VLN does not occur.

5. Between two VLN clients S and R in the same Cronus cluster, the sequence of datagrams received by R is a subsequence of the sequence sent by S to R; a stronger sequencing property holds for broadcast and multicast addressing.

2.1 VLN Addressing

In the DARPA internet an 'internet address' is defined to be a 32 bit quantity which is partitioned into two fields, a network number and a 'local address'. VLN addresses share this basic structure, and are perceived by hosts outside the Cronus system as ordinary internet addresses. A sender outside a Cronus cluster may direct an internet datagram into the cluster by specifying the VLN network number in the network number field of the destination address; senders in the cluster may transmit messages to internet hosts outside the cluster in a similar way. The VLN in a Cronus cluster, however, attaches special meaning to the local address field of a VLN address, as explained below.

Each network in the internet community is assigned a 'class', either A, B, or C, and a network number in its class. The partitioning of the 32 bit internet address into network number and local address fields is a function of the class of the network number, as follows:

	Width of Network Number	Width of Local Address
Class A	7 bits	24 bits
Class B	14 bits	16 bits
Class C	21 bits	8 bits

Table 1. Internet Address Formats

The bits not included in the network number or local address fields encode the network class, e.g., a 3 bit prefix of 110 designates a class C address (see [4]).

The interpretation of the local address field of an internet address is the responsibility of the network designated in the network number field. In the ARPANET (a class A network, with network number 10) the local address refers to a specific physical host; this is the most common use of the local address field. VLN addresses, in contrast, may refer to all hosts (broadcast) or groups of hosts (multicast) in a Cronus cluster, as well as specific hosts inside or outside of the Cluster. Specific, broadcast, and multicast addresses are all encoded in the VLN local address field. (4)

The meaning of the local address field of a VLN address is defined in the table below.

ADDRESS MODES	VLN LOCAL ADDRESS VALUES
Specific Host	0 to 1,023
Multicast	1,024 to 65,534
Broadcast	65,535

Table 2. VLN Local Address Modes

In order to represent the full range of specific, broadcast, and multicast addresses in the local address field, a VLN network should be either class A or class B. If a VLN is a class A internet network, a VLN local address occupies the low-order 16 bits of the 24 bit internet local address field, and the upper 8 bits of the internet local address are zero. If a VLN is a class

---

(4) The ability of hosts outside a Cronus cluster to transmit datagrams with VLN broadcast or multicast destination addresses into the cluster may be restricted by the cluster gateway(s), for reasons of system security.

B network, the internet local address field is fully utilized by the VLN local address.

2.2 VLN Operations

There are seven operations defined at the VLN interface and available to the VLN client on each host. An implementation of the VLN interface has wide lattitude in the presentation of these operations to the client; for example, the operations may or may not return error codes.

A VLN implementation may define the operations to occur synchronously or asynchronously with respect to the client's computation. We expect that the ResetVLNInterface, MyVLNAddress, SendVLNDatagram, PurgeMAddresses, AttendMAddress, and IgnoreMAddress operations will usually be synchronous with respect to the client, but ReceiveVLNDatagram will usually be asynchronous, i.e., the client may initiate the operation, continue to compute, and at some later time be notified that a datagram is available. (The alternatives to asynchronous

ReceiveVLNDatagram are A) a blocking receive operation; and B) a non-blocking but synchronous receive operation, which returns a failure code immediately if a datagram is not available. Either alternative may satisfy particular requirements, but an asynchronous receive subsumes these and is more generally useful.) At a minimum, the client must have fully synchronous access to each of the operations; more elaborate mechanisms may be provided at the option of the VLN implementation.

VLN OPERATIONS

ResetVLNInterface

The VLN layer for this host is reset (e.g., for the Ethernet VLN implementation the operation ClearVPMap is performed, and a frame of type "Cronus VLN" and subtype "Mapping Update" is broadcast; see Section 4.2). This operation does not affect the set of attended VLN multicast addresses.

function MyVLNAddress()

Returns the specific VLN address of this host; this can always be done without communication with any other host.

SendVLNDatagram(Datagram)

When this operation completes, the VLN layer has copied the Datagram and it is either "in transmission" or "delivered", i.e., the transmitting process cannot assume that the message has been delivered when SendVLNDatagram completes.

ReceiveVLNDatagram(Datagram)

When this operation completes, Datagram is a representation of a VLN datagram sent by a VLN client and not previously received by the client invoking ReceiveVLNDatagram.

PurgeMAddresses()

When this operation completes, no VLN multicast addresses are registered with the local VLN component.

function AttendMAddress(MAddress)

If this operation returns True then MAddress, which must be a VLN multicast address, is registered as an "alias" for this host, and messages addressed to MAddress by VLN clients will be delivered to the client on this host.

IgnoreMAddress(MAddress)

When this operation completes, MAddress is not registered as a multicast address for the client on this host.

Whenever a Cronus host comes up, ResetVLNInterface and PurgeMAddresses are performed implicitly by the VLN layer before it will accept a request from the client or incoming traffic from the PLN. They may also be invoked by the client during normal operation. As described in Section 4.2 below, a VLN component may depend upon state information obtained dynamically from other hosts, and there is a possibility that incorrect information might enter a component's state tables. (This might happen, for example, if the PLN address of a Cronus host were changed but its VLN address preserved--the old VLN-to-PLN address mappings held by other hosts would then be incorrect.) A cautious VLN client could call ResetVLNInterface at periodic intervals (every hour, say) to force the VLN component to reconstitute its dynamic tables.

A VLN component will place a limit on the number of multicast addresses to which it will simultaneously "attend"; if the client attempts to register more addresses than this, AttendMAddress will return False with no other effect. The actual limit will vary among VLN components, but it will usually be between 10 and 100 multicast addresses. Components may implement limits as large as the entire multicast address space (64,511 addresses).

The VLN layer does not guarantee any minimum amount of buffering for datagrams, at either the sending or receiving host(s). It does guarantee, however, that a SendVLNDatagram operation invoked by a VLN client will eventually complete; this implies that datagrams may be lost if buffering is insufficient and receiving clients are too slow. The VLN layer will do its best to discard packets for this reason very infrequently.

2.3 Reliability Guarantees

Guarantees are never absolute--there is always some probability, however remote, that a catastrophe will occur and a promise be broken. Nevertheless, the concept of a guarantee is still valuable, because the improbability of a catastrophic failure influences the design and cost of the recovery mechanisms needed to overcome it. In this spirit, the word "guarantee" as used here implies only that the alternatives to correct function (i.e., catastrophic failures) are extremely rare events.

The VLN does not attempt to guarantee reliable delivery of datagrams, nor does it provide negative acknowlegements of damaged or discarded datagrams. It does guarantee that received datagrams are accurate representations of transmitted datagrams.

The VLN also guarantees that datagrams will not "replicate" during transmission, i.e., for each intended receiver, a given datagram is received once or not at all. (5)

Between two VLN clients S and R in the same cluster, the sequence of datagrams received by R is a subsequence of the sequence sent by S to R, i.e., datagrams are received in order, possibly with omissions.

A stronger sequencing property holds for broadcast and multicast transmissions. If receivers R1 and R2 both receive broadcast or multicast datagrams D1 and D2, either they both receive D1 before D2, or they both receive D2 before D1.

3 Desirable Characteristics of a Physical Local Network

While it is conceivable that a VLN could be implemented on a long-haul or virtual-circuit-oriented PLN, these networks are generally ill-suited to the task. The ARPANET, for example, does not support broadcast or multicast addressing modes, nor does it provide the VLN sequencing guarantees. If the ARPANET were the base for a VLN implementation, broadcast and multicast would have to be constructed from specific addressing, and a network-wide synchronization mechanism would be required to implement the sequencing guarantees. Although the compatibility and substitutability benefits might still be achieved, the implementation would be costly, and performance poor.

---

(5) A protocol operating above the VLN layer (e.g., TCP) may employ a retransmission strategy; the VLN layer does nothing to filter duplicates arising in this way.

A good implementation base for a Cronus VLN would be a high-bandwidth local network with all or most of these characteristics:

1. The ability to encapsulate a VLN datagram in a single PLN datagram.
2. An efficient broadcast addressing mode.
3. Natural resistance to datagram replication during transmission.
4. Sequencing guarantees like those of the VLN interface.
5. A strong error-detecting code (datagram checksum).

Good candidates include Ethernet, the Flexible Intraconnect, and Pronet, among others.

4 A VLN Implementation Based on Ethernet

The Ethernet local network specification is the result of a collaborative effort by Digital Equipment Corp., Intel Corp., and Xerox Corp. The Version 1.0 specification [3] was released in September, 1980. Useful background information on the Ethernet internetworking model is supplied in [2].

The Ethernet VLN implementation begins with the assumption, in accordance with the model developed in [2], that the addresses of specific Ethernet hosts are arbitrary, 48 bit quantities, not under the control of DOS Design/Implementation Project. The VLN implementation must, therefore, develop a strategy to map VLN addresses to specific Ethernet addresses.

A second important assumption is that the VLN-address-to-Ethernet-address mapping should not be maintained manually in each VLN host. Manual procedures are too cumbersome and error-prone when a local network may consist of hundreds of hosts, and hosts may join and leave the network frequently. A protocol is described below which allows hosts to dynamically construct the mapping, beginning only with knowledge of their own VLN and

Ethernet host addresses.

The succeeding sections discuss the VLN implementation based on the Ethernet PLN in detail, as designed for the Cronus prototype currently being assembled by Bolt Beranek and Newman, Inc.

4.1 Datagram Encapsulation

An internet datagram is encapsulated in an Ethernet frame by placing the internet datagram in the Ethernet frame data field, and setting the Ethernet type field to "DoD IP".

To guarantee agreement by the sending and receiving VLN components on the ordering of internet datagram octets within an encapsulating Ethernet frame, the Ethernet octet ordering is required to be consistent with the IP octet ordering. Specifically, if IP(i) and IP(j) are internet datagram octets and i<j, and EF(k) and EF(l) are the Ethernet frame octets which represent IP(i) and IP(j) once encapsulated, then k<l. Bit

orderings within octets must also be consistent. (6)

4.2 VLN Specific Addressing Mode

Each VLN component maintains a virtual-to-physical address map (the VPMap) which translates a 32 bit specific VLN host address (7) in this cluster to a 48 bit Ethernet address. (8) The VPMap data structure and the operations on it can be efficiently implemented using standard hashing techniques. Only three operations defined on the VPMap are discussed in this note: ClearVPMap, TranslateVtoP, and StoreVPPair.

Each host has an Ethernet host address (EHA) to which its controller will respond, determined by Xerox and the controller manufacturer (see Section 4.5.2). At host initialization time,

---

(6) See [1] for a lively discussion of the problems arising from the failure of communicants to agree upon consistent orderings.

(7) Since the high-order 22 bits of the address are constant for all specific host addresses in a cluster, only the low-order 10 bits of the address are significant.

(8) The least significant bit of the first octet of the Ethernet address is always 0, since these are not broadcast or multicast addresses.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination Address (contd.)  |         Source Address        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Source Address (contd.)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type  ("DoD IP")           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |Version|  IHL  |Type of Service|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Total Length           |         Identification        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Flags|     Fragment Offset     |  Time to Live |    Protocol   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Header Checksum          |          Source Address       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Source Address (contd.)     |      Destination Address      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination Address (contd.)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
.                                                               .
.                             Data                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Frame Check Sequence                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Table 3. An Encapsulated Internet Datagram

the local VLN component establishes a second host address, the multicast host address (MHA), constructed from the host's VLN address. Represented as a sequence of octets in hexadecimal, the MHA has the form:

```
     A  B  C  D  E  F

    09-00-08-00-hh-hh
```

A is the first octet transmitted, and F the last. The two octets E and F contain the host local address:

```
       E        F

    000000hh  hhhhhhhh
          ↑          ↑
         MSB        LSB
```

When the VLN client invokes SendVLNDatagram to send a specifically addressed datagram, the local VLN component encapsulates the datagram in an Ethernet frame and transmits it without delay. The Source Address in the Ethernet frame is the EHA of the sending host. The Ethernet Destination Address is formed from the destination VLN address in the datagram, and is either:

- the EHA of the destination host, if the TranslateVtoP operation on the VPMap succeeds,

or

- the MHA formed from the host number in the destination VLN address, as described above.

When a VLN component receives an Ethernet frame with type "DoD IP", it decapsulates the internet datagram and delivers it to its client. If the frame was addressed to the EHA of the receiving host, no further action is taken, but if the frame was addressed to the MHA of the receiving host the VLN component will broadcast an update for the VPMaps of the other hosts. This will permit the other hosts to use the EHA of this host for future traffic. The type field of the Ethernet frame containing the update is "Cronus VLN", and the format of the data octets in the frame is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Subtype ("Mapping Update")  |        Host VLN Address       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Host VLN Address (contd.)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

When a local VLN component receives an Ethernet frame with type

"Cronus VLN" and subtype "Mapping Update", it performs a StoreVPPair operation using the Ethernet Source Address field and the host VLN address sent as frame data.

This multicast mechanism could be extended to perform other address mapping functions, for example, to discover the addresses of a cluster's gateways. Suppose all gateways register the same Multicast Gateway Address (MGA, analogous to MHA) with their Ethernet controllers; the MGA then becomes a "logical name" for the gateway function in a Cronus cluster. If a host needs to send a datagram out of the cluster and doesn't know what specific gateway address to use, the host can multicast the datagram to all gateways by sending to MGA. One or more of the gateways can forward the datagram, and transmit a "Gateway Mapping Update" (containing the gateway's specific Ethernet address) back to the originating host. Specific gateway addresses could be cached in a structure similar to the VPMap, keyed to the destination network number. (9)

---

(9) Because the Cronus Advanced Development Model will contain only one gateway, a simpler mechanism will be implemented initially; the specific Ethernet address of the gateway will be "well-known" to all VLN components.

The approach just outlined suggests that all knowledge of the existence and connectivity of gateways would be isolated in the VLN layer of cluster hosts. Other mechanisms, e.g., based on the ICMP component of the Internet Protocol, could be used instead to disseminate information about gateways to cluster hosts (see [7]). These would require, however, specific Ethernet addresses to be visible above the VLN layer, a situation the current design avoids.

4.3 VLN Broadcast and Multicast Addressing Modes

A VLN datagram will be transmitted in broadcast mode if the argument to SendVLNDatagram specifies the VLN broadcast address (local address = 65,535, decimal) as the destination. Broadcast is implemented in the most straightforward way: the VLN datagram is encapsulated in an Ethernet frame with type "DoD IP", and the frame destination address is set to the Ethernet broadcast address. The receiving VLN component merely decapsulates and delivers the VLN datagram.

The implementation of the VLN multicast addressing mode is more complex, for several reasons. Typically, each VLN host will define a constant called Max←Attended, equal to the maximum number of VLN multicast addresses which can be simultaneously "attended" by this host. Max←Attended should not be a function of the particular Ethernet controller(s) the host may be using, but only of the software resources (buffer space and processor time) that the host dedicates to VLN multicast processing. The protocol below permits a host to attend any number of VLN multicast addresses, from 0 to 64,511 (the entire VLN multicast address space), independent of the controller in use.

Understanding of the VLN multicast protocol requires some knowledge of the behavior of existing Ethernet controllers. The Ethernet specification does not specify whether a controller must perform multicast address recognition, or if it does, how many multicast addresses it must be prepared to recognize. As a result Ethernet controller designs vary widely in their behavior. For example, the 3COM Model 3C400 controller follows the first pattern and performs no multicast address recognition, instead passing all multicast frames to the host for further processing.

The Intel Model iSBC 550 controller permits the host to register a maximum of 8 multicast addresses with the controller, and the Interlan Model NM10 controller permits a maximum of 63 registered addresses.

It would be possible to implement the VLN multicast mode using only the Ethernet broadcast mechanism. This would imply, however, that every VLN host would receive and process every VLN multicast, often only to discard the datagram because it is misaddressed. More efficient operation is possible if at least some Ethernet multicast addresses are used, since Ethernet controllers with multicast recognition can discard misaddressed frames more rapidly than their hosts, reducing both the processor time and buffer space demands upon the host.

The protocol specified below satisfies the design constraints and is especially simple.

A VLN-wide constant, Min←Attendable, is equal to the smallest number of Ethernet multicast addresses that can be simultaneously attended by any host in the VLN, or 64,511, whichever is smaller. A network composed of hosts with the Intel and Interlan controllers mentioned above, for example, would have Min←Attendable equal to 7; (10) a network composed only of hosts with 3COM Model 3C400 controllers would have Min←Attendable equal to 64,511, since the controller itself does not restrict the number of Ethernet multicast addresses to which a host may attend. (11)

The local address field of a VLN multicast address can be represented in two octets, in hexadecimal:

```
mm-mm
```

From Table 1, mm-mm considered as a decimal integer M is in the range 1,024 to 65,534. When SendVLNDatagram is invoked with a VLN multicast datagram, there are two cases:

1. (M - 1,023) <= Min←Attendable. In this case, the datagram is encapsulated in a "DoD IP" Ethernet frame, and multicast with the Ethernet address

   ```
   09-00-08-00-mm-mm
   ```

   A VLN component which attends VLN multicast addresses in this range should receive Ethernet multicast addresses in this format, if necessary by registering the addresses with its Ethernet controller.

---

(10) Min←Attendable is 7, rather than 8, because one multicast slot in the controller must be reserved for the host's MHA, as described in Section 4.2.

(11) For the Cronus Advanced Development Model, Min←Attendable is currently defined to be 60.

2. (M - 1,023) > Min←Attendable. The datagram is encapsulated in a "DoD IP" Ethernet frame, and transmitted to the Ethernet broadcast address. A VLN component which attends VLN multicast addresses in this range must receive all broadcast frames, and filter them on the basis of frame type and VLN destination address (found in the IP destination address field).

There are two drawbacks to this protocol that might induce a more complex design: 1) because Min←Attendable is the "lowest common denominator" for the ability of Ethernet controllers to recognize multicast addresses, some controller capabilities may be wasted; 2) small VLN addresses (less than Max←Attendable + 1,024) will probably be handled more efficiently than large VLN multicast addresses. The second factor complicates the assignment of VLN multicast addresses to functions, since the particular assignment affects multicast performance.

4.4 Reliability Guarantees

Delivered datagrams are accurate copies of transmitted datagrams because VLN components do not deliver incoming datagrams with invalid Frame Check Sequences. The 32 bit CRC error detecting code applied to Ethernet frames is very powerful, and the probability of an undetected error occuring "on the wire" is very small. The probability of an error being introduced before the checksum is computed or after it is checked is comparable to the probability of an error in a disk subsystem before a write operation or after a read; often, but not always, it can be ignored.

Datagram duplication does not occur because the VLN layer does not perform datagram retransmissions, the primary source of duplicates in other networks. Ethernet controllers do perform retransmission as a result of "collisions" on the channel, but the "collision enforcement" or "jam" assures that no controller receives a valid frame if a collision occurs.

The sequencing guarantees hold because mutually exclusive access to the transmission medium defines a total ordering on

Ethernet transmissions, and because a VLN component buffers all datagrams in FIFO order, if it buffers more than one datagram.

4.5 Use of Assigned Numbers

On a philosophical note, protocols such as IP and TCP exist to provide communication services to extensible sets of clients; new clients and usages continue to emerge over the life of a protocol. Because a protocol implementation must have some unambiguous knowledge of the "names" of the clients, sockets, hosts, networks, etc., with which it interacts, a need arises for the continuing administration of the 'assigned numbers' related to the protocol. Typically the organization which declares a protocol to be a standard also becomes the administrator for its assigned numbers. The organization will designate an office to assign numbers to the clients, sockets, hosts, networks, etc., that emerge over time. The office will also prepare lists of number assignments that are distributed to protocol users; the reference [4] is a list of this kind.

There are three organizations responsible for number assignment related to the Ethernet-based VLN implementation: DARPA, Xerox, and the DOS Design/Implementation Project; their respective roles are described below.

4.5.1 DARPA

DARPA administers the internet network number and internet protocol number assignments. The Ethernet-based VLN implementation does not involve DARPA assigned numbers, but any particular 'instance' of a Cronus VLN is expected to have a class A or B internet network number assigned by DARPA. For example, the prototype Cronus system (the Advanced Development Model) being constructed at Bolt Beranek and Newman, Inc., has class B network number 128.011.xxx.xxx.

Protocols built above the VLN will make use of other DARPA assigned numbers, e.g., the Cronus object-operation protocol requires an internet protocol number.

4.5.2 The Xerox Ethernet Address Administration Office

The Ethernet Address Administration Office at Xerox Corp. administers Ethernet specific and multicast address assignments, and Ethernet frame type assignments.

It is the intent of the Xerox internetworking model that every Ethernet host have a distinct specific address, and that the address space be large enough to accomodate a very large population of inexpensive hosts (e.g., personal workstations). They have therefore chosen to delegate the authority to assign specific addresses to the manufacturers of Ethernet controllers, by granting them large blocks of addresses on request. Manufacturers are expected to assign specific addresses from these blocks densely, e.g., sequentially, one per controller, and to consume all of them before requesting another block.

The preceding paragraph explains the Xerox address assignment policy not because the DOS Design/Implementation Project intends to manufacture Ethernet controllers (!), but because Xerox has chosen to couple the assignment of specific and multicast Ethernet addresses. An assigned block is defined by a

23-bit constant, which specifies the contents of the first three octets of an Ethernet address, except for the broadcast/multicast bit (the least significant bit of the first octet). The possessor of an assigned block thus has in hand 2**24 specific addresses and 2**24 multicast addresses, to parcel out as necessary.

The block assigned for use in the Cronus system is defined by the octets 08-00-08 (hex). The specific addresses in this block range from 08-00-08-00-00-00 to 08-00-08-FF-FF-FF (hex), and the multicast addresses range from 09-00-08-00-00-00 to 09-00-08-FF-FF-FF (hex). Only a fraction of the multicast addresses are actually utilized, as explained in Sections 4.2 and 4.3.

The Ethernet Address Administration Office has designated a public frame type, "DoD IP", 08-00 (hex), to be used for encapsulated internet protocol datagrams. The Ethernet VLN implementation uses this frame type exclusively for datagram encapsulation. In addition, the Cronus system uses two private Ethernet frame types, assigned by the Ethernet Address Administration Office:

NAME	TYPE
Cronus VLN	80-03
Cronus Direct	80-04

(The use of the "Cronus Direct" frame type is not described in this note.)

The same Ethernet address and frame type assignments will be used by every instance of a Cronus VLN; no further assignments from the Ethernet Address Administration Office are anticipated.

4.5.3 The DOS Design/Implementation Project

The DOS Design/Implementation Project assumes responsibility for the assignment of subtypes of the Ethernet frame type "Cronus VLN". No assignments of subtypes for purposes unrelated to the Cronus system design are expected, nor are assignments to other organizations. The subtypes currently assigned are:

NAME	SUBTYPE
Mapping Update	00-01

REFERENCES

[1]
"On holy wars and a plea for peace," Danny Cohen, Computer, V 14 N 10, October 1981, pp. 48-54.

[2]
"48-bit absolute internet and Ethernet host numbers," Yogen K. Dalal and Robert S. Printis, Proc. of the 7th Data Communications Symposium, October 1981.

[3]
"The Ethernet: a local area network, data link layer and physical layer specifications," Digital Equipment Corp., Intel Corp., and Xerox Corp., Version 1.0, September 1980.

[4]
"Assigned numbers," Jon Postel, RFC 790, USC/Information Sciences Institute, September 1981.

[5]
"Internet Protocol - DARPA internet program protocol specification," Jon Postel, ed., RFC 791, USC/Information Sciences Institute, September 1981.

[6]
"Internet protocol transition workbook," Network Information Center, SRI International, Menlo Park, California, March 1982.

[7]
"IP - Local Area Network Addressing Issues," Robert Gurwitz and Robert Hinden, Bolt Beranek and Newman Inc., (draft) August 1982.